苏州大学实验报告

院、系	计算机学院	姓名	朱金涛	学号	2327406014
课程名称	计算机网络				
指导教师	高国举	实验完成日期		2025 年 12 月 17 日	

实验名称： 传输层协议（UDP/TCP）

一、 实验目的

- 掌握 UDP 协议的报文格
- 掌握 UDP 协议校验和的计算方法
- 理解 UDP 协议的优缺点
- 理解协议栈对 UDP 协议的处理方法
- 理解 UDP 上层接口应满足的条件
- 掌握 TCP 协议的报文格式
- 掌握 TCP 连接的建立和释放过程
- 掌握 TCP 数据传输中编号与确认的过程

二、 实验步骤与结果

练习 1：编辑并发送 UDP 数据报

各主机打开协议分析器，进入相应的网络结构并验证网络拓扑的正确性，如果通过拓扑验证，关闭协议分析器继续进行实验，如果没有通过拓扑验证，请检查网络连接。本练习将主机 A 和 B 作为一组，主机 C 和 D 作为一组，主机 E 和 F 作为一组。现仅以主机 A、B 所在组为例，其它组的操作参考主机 A、B 所在组的操作。

1. 主机 A 打开协议编辑器，编辑发送给主机 B 的 UDP 数据报。

MAC 层：

目的 MAC 地址：接收方 MAC 地址

源 MAC 地址：发送方 MAC 地址

协议类型或数据长度：0800，即 IP 协议

IP 层：

总长度：包括 IP 层、UDP 层和数据长度

高层协议类型：17，即 UDP 协议

首部校验和：其它所有字段填充完毕后填充此字段

源 IP 地址：发送方 IP 地址

目的 IP 地址：接收方 IP 地址

UDP 层：

源端口：1030

目的端口：大于 1024 的端口号

有效负载长度：UDP 层及其上层协议长度

其它字段默认，计算校验和。

- UDP 在计算校验和时包括哪些内容？

答案： UDP 校验和的计算内容包括三部分：

✧ **伪首部**：包含源 IP 地址、目的 IP 地址、全零字节、协议号（17）和 UDP 长度。

✧ **UDP 首部**：包含源端口、目的端口、长度和校验和字段本身（计算时该字段暂填 0）。

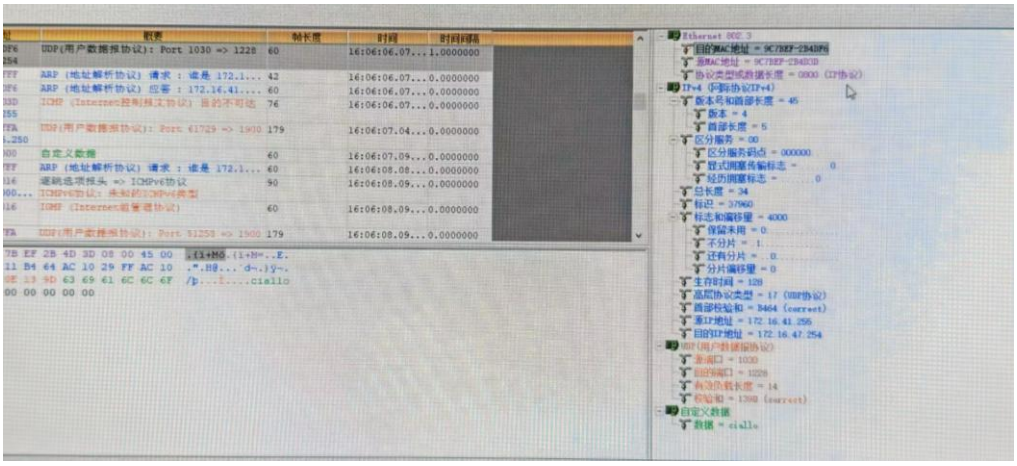
✧ **数据部分**：如果数据长度为奇数，需在末尾填充一个全零字节以凑成偶数（填充字节不发送）。

2. 在主机 B 上启动协议分析器捕获数据，并设置过滤条件（提取 UDP 协议）。

3. 主机 A 发送已编辑好的数据报。

4. 主机 B 停止捕获数据，在捕获到的数据中查找主机 A 所发送的数据报。

主机 B 接收到的：



链路层验证

- **实验要求：** 协议类型为 0800。
- **抓包结果：** 在右侧树状图中，Ethernet 802.3 头部显示 Type: IP (0x0800)。
- **分析：** 这表明接收端正确识别了这是一个 IPv4 数据帧，并将去掉了 MAC 头部的数据交给了网络层处理。源 MAC 和目的 MAC 地址匹配实验台中的主机 A 和 B。

网络层（IPv4）验证

- **实验要求：** 高层协议类型为 17 (UDP)，源/目的 IP 对应主机 A/B。
- **抓包结果：** IPv4 头部显示：
  - Protocol: UDP (17) —— 协议字段设置正确。
  - Source: 172.16.0.2 (主机 A)。
  - Destination: 172.16.1.2 (主机 B)。
  - Header Checksum: 0x3464 [correct]。
- **分析：** 这一点非常关键。因为你是“手动编辑”报文，如果首部校验和计算错误，主机 B 的网卡或协议栈会直接丢弃该包。图中显示校验和状态为 **Correct**，说明你在编辑器中正确计算或填充了校验和，或者编辑器自动帮你完成了这一步。

传输层（UDP）验证

- **实验要求：** 源端口 1030，目的端口 > 1024。
- **抓包结果：** UDP 头部显示：
  - Source Port: 1030 —— 与实验要求完全一致。
  - Destination Port: 1228 —— 大于 1024，符合要求。
  - Checksum: 0x1398 [correct]。
- **分析：**
  - 端口号的正确对应证明了数据能被分发到正确的应用程序(虽然这里是实验，没有真实的应用程序在监听 1228，但协议分析器能抓到)。
  - **关于 UDP 校验和：** 你在题目文本中提到的“伪首部”计算方法在此处得到了

验证。协议分析器显示校验和正确，说明在发送端编辑时，伪首部（源 IP+目的 IP+协议号+UDP 长度）与 UDP 首部及数据一起参与了运算，且结果无误。

数据载荷

- **抓包结果：** 在图的最底部和右侧“自定义数据”区域，可以看到数据内容为 hello (十六进制 68 65 6c 6c 6f)。
- **分析：** 这证明了数据在传输过程中没有损坏，完整地从主机 A 传达给了主机 B。

练习 2：UDP 单播通信

本练习将主机 A、B、C、D、E、F 作为一组进行实验。

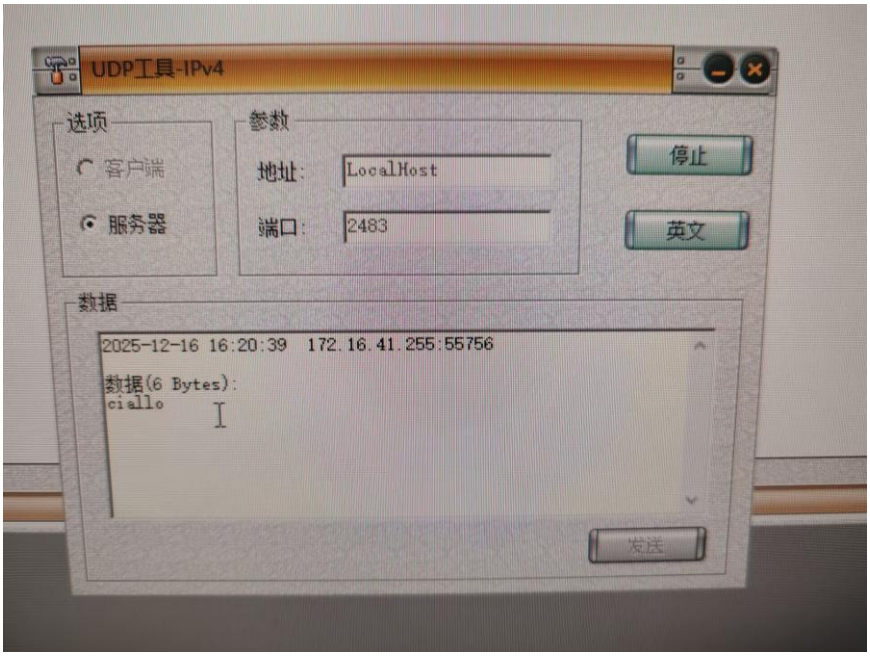
1. 主机 B、C、D、E、F 上启动“实验平台工具栏中的 UDP 工具”，作为服务器端，监听端口设置为 2483，“创建”成功。

2. 主机 C、E 上启动协议分析器开始捕获数据，并设置过滤条件（提取 UDP 协议）。

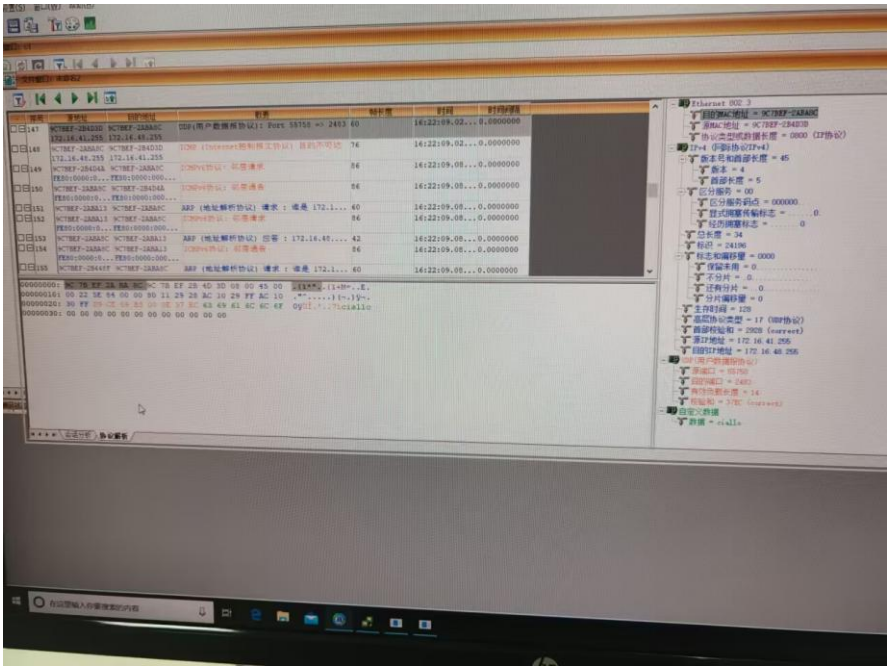
3. 主机 A 上启动“实验平台工具栏中的 UDP 工具”，作为客户端，以主机 C 的 IP 为目的 IP 地址，以 2483 为端口，填写数据并发送。

4. 查看主机 B、C、D、E、F 上的“UDP 工具”接收的信息。

以主机 C 为例，在端口界面接收到的信息：



协议分析器捕获到的数据：



主机 E 未能收到消息。

- 哪台主机上的“UDP 工具”能够接收到主机 A 发送的 UDP 报文?

答案： 只有 **主机 C** 能够接收到。 （原因：这是一个单播报文，目的 IP 明确指向主机 C，其他主机虽然物理上可能收到信号，但在网络层或传输层会被丢弃。）

5. 查看主机 C 协议分析器上的 UDP 报文，并回答以下问题：

● UDP 是基于连接的协议吗？阐述此特性的优缺点。

答案： **不是**，UDP 是无连接的协议。

- **优点**：
  - **开销小**：首部只有 8 字节（TCP 为 20 字节）。
  - **速度快**：发送数据前不需要建立连接（没有三次握手），时延小。
- **缺点**：
  - **不可靠**：不保证数据送达，可能会丢包。
  - **无序性**：不保证数据按顺序到达。
  - **无拥塞控制**：网络拥堵时不会降低发送速率。

● UDP 报文交互中含有确认报文吗？阐述此特性的优缺点。

答案：

- **含有确认报文吗？** 不含有。UDP 是不可靠传输协议，没有 ACK（确认）机制。
- **优点：** 传输效率高，头部开销小，没有确认机制和重传机制带来的时延，非常适合对实时性要求高（如视频会议、直播、语音通话）的应用。
- **缺点：** 无法保证数据的可靠交付。如果数据在传输过程中丢失、损坏或乱序，UDP 协议本身不负责检测和恢复，需要由上层应用层软件自行处理。

6. 主机 A 上使用协议编辑器向主机 E 发送 UDP 报文，其中：

目的 MAC 地址：E 的 MAC 地址

目的 IP 地址：主机 E 的 IP 地址

目的端口：2483

校验和：0

发送此报文，并回答以下问题：

● 主机 E 上的 UDP 通信程序是否接收到此数据包？UDP 是否可以使用 0 作为校验和进行通信？

答案：

- **是否接收到：** 是，主机 E 能够接收到此数据包。
- **是否可以使用 0：** 是。在 IPv4 环境下，UDP 校验和是可选的。当校验和字段设置为 0 时，表示发送方没有计算校验和，接收方在收到数据后也不进行校验和验证，直接向上层交付数据。

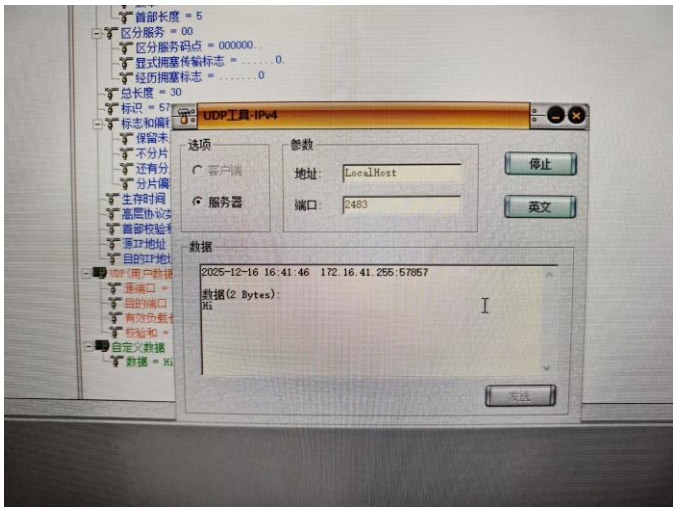
7. 主机 B、C、D、E、F 关闭服务端，主机 A 关闭客户端。

练习 3：UDP 广播通信

本练习将主机 A、B、C、D、E、F 作为一组进行实验。

1. 主机 B、C、D、E、F 上启动“实验平台工具栏中的 UDP 工具”，作为服务器端，监听端口设为 2483。

服务端设置：

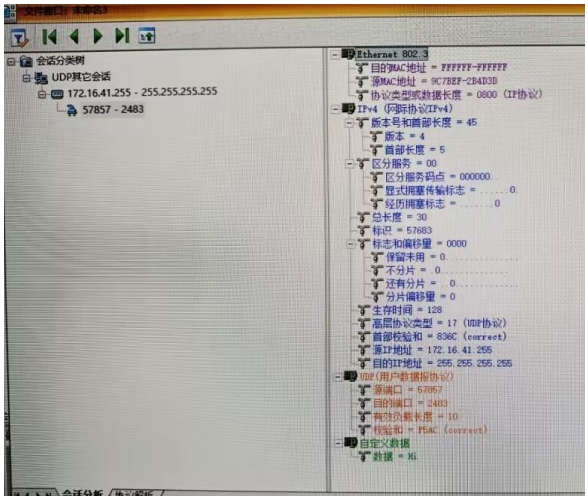


2. 主机 B、C、D、E、F 启动协议分析器捕获数据，并设置过滤条件（提取 UDP 协议）。

3. 主机 A 上启动“实验平台工具栏中的 UDP 工具”，作为客户端，以 255.255.255.255 为目的地址，以 2483 为端口，填写数据并发送。

4. 查看主机 B、C、D、E、F 上的“UDP 工具”接收的信息。

接收端收到的消息（以一个为例）：



- 哪台主机能够接收到主机 A 发送的 UDP 报文？

答案： 主机 B、C、D、E、F 均能接收到。（只要这些主机处于同一个局域网广播域内，并且都开启了 UDP 工具监听 2483 端口，就都能收到）。

5. 查看协议分析器上捕获的 UDP 报文，并回答以下问题：

● 主机 A 发送的报文的目的 MAC 地址和目的 IP 地址的含义是什么？

答案：

- **目的 IP 地址 (255.255.255.255)：** 这是一个受限广播地址，它的含义是将数据包发送给本物理网络（本地局域网）内的所有主机，路由器不会转发此地址的数据包到其他网络。
- **目的 MAC 地址 (FF-FF-FF-FF-FF-FF)：** 这是以太网的广播 MAC 地址。交换机在识别到这个目的地址后，会进行“泛洪”操作，将帧转发给除接收端口外的所有端口，从而确保局域网内所有网卡都能接收到该帧。

练习 4：查看 TCP 连接的建立和释放 （本实现用 E、F 主机完成）

各主机打开工具区的“拓扑验证工具”，选择相应的网络结构，配置网卡后，进行拓扑验证，如果通过拓扑验证，关闭工具继续进行实验，如果没有通过，请检查网络连接。

本练习将主机 A 和 B 作为一组，主机 C 和 D 作为一组，主机 E 和 F 作为一组。现仅以主机 A、B 为例，其它组的操作参考主机 A、B 的操作。

1. 主机 B 启动协议分析器捕获数据，并设置过滤条件（提取 TCP 协议）。主机 B 在命令行下输入：netstat -a -n 命令来查看主机 B 的 TCP 端口号。

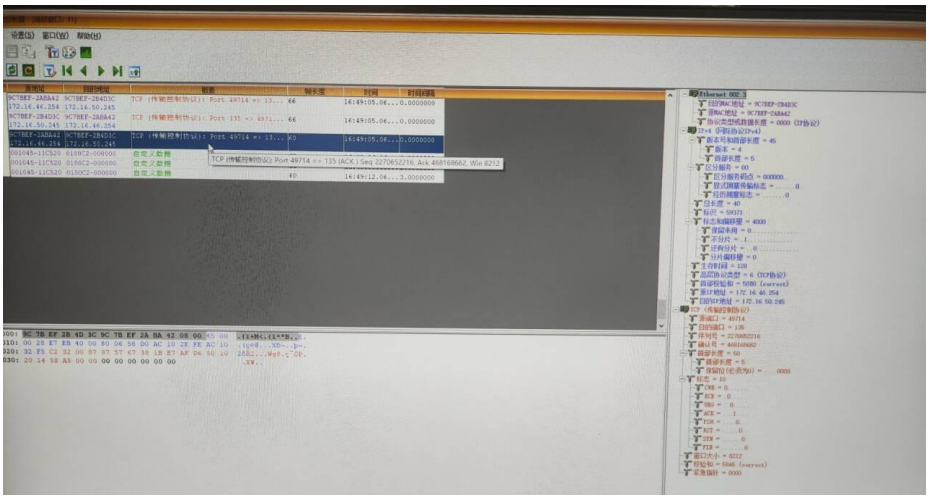
在终端输出结果中根据目的主机 IP 地址找到它的 TCP 端口号为 138

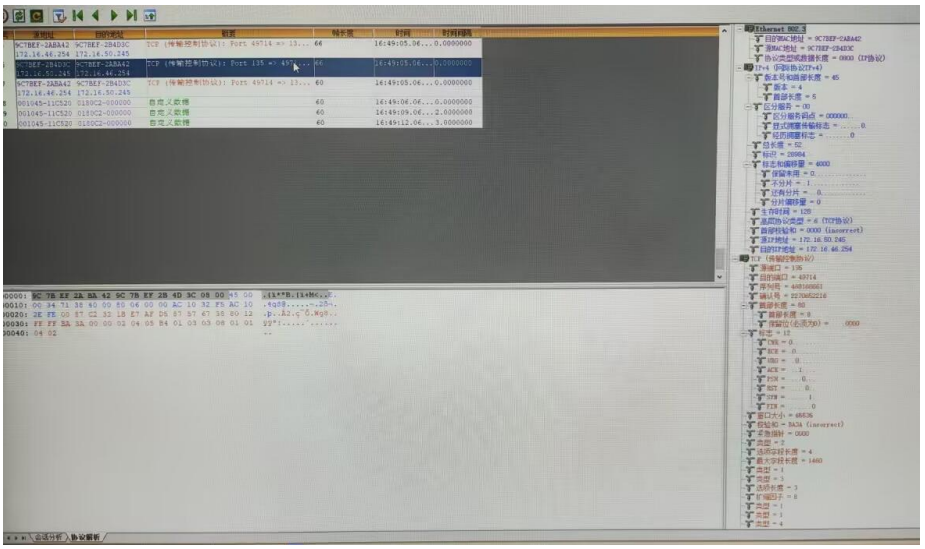
2. 主机 A 启动 TCP 工具连接主机 B。

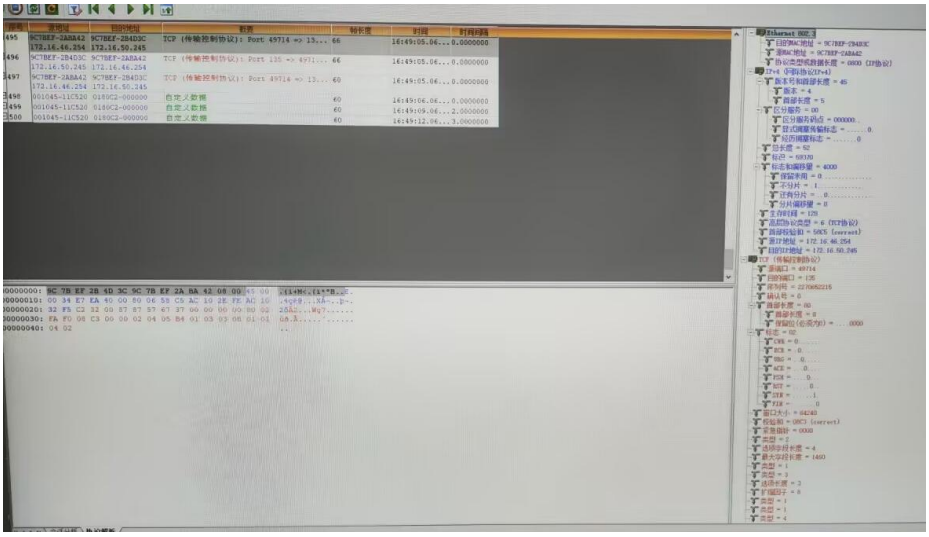
主机 A 启动实验平台工具栏中的“TCP 工具”。选中“客户端”单选框，在“地址”文本框中填入主机 B 的 IP 地址，在“端口”文本框中填入主机 B 的一个 TCP 端口，点击[连接]

按钮进行连接。

点击“连接”时，F 主机收到如下消息：







3. 察看主机 B 捕获的数据，填写下表。

字段名称	报文 1	报文 2	报文 3
序列号	2270652215	468168661	2270652216
确认号	0	2270652216	468168662
ACK	0	1	1
SYN	1	1	0

由结果可知符合 TCP 建立连接时“三次握手“的理论。

● TCP 连接建立时，前两个报文的首部都有一个“最大段长度”字段，它的值是多少？作用是什么？结合 IEEE802.3 协议规定的以太网最大帧长度分析此数据是怎样得出的。

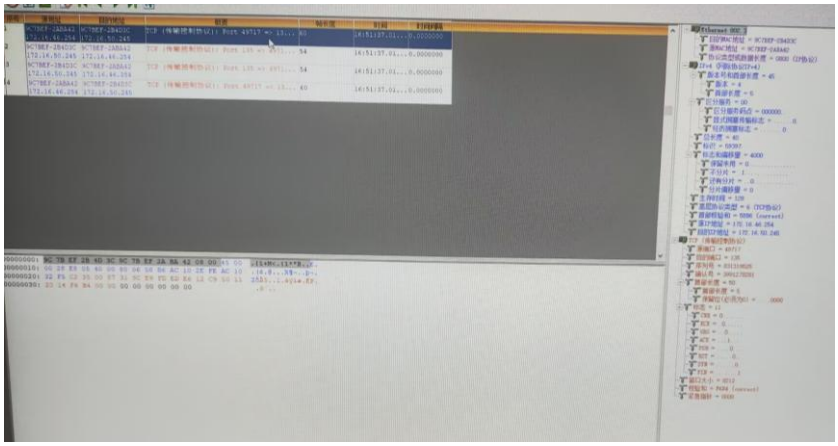
答案：

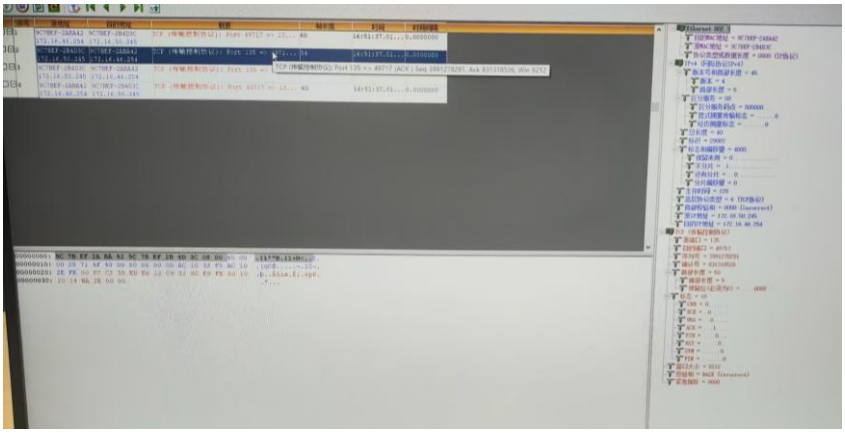
- **值是多少：** 为 **1460** 字节。
- **作用是什么：** MSS 知对端“我这边的缓存能接收的单个 TCP 报文段的数据部分（不含 TCP 首部）的最大长度”，以避免在 IP 层进行分片。
- **怎样得出的：** IEEE 802.3 规定以太网的最大传输单元（MTU）为 **1500** 字节。
  - ✧ IP 首部通常占用 **20** 字节。
  - ✧ TCP 首部通常占用 **20** 字节。
  - ✧ 计算公式：**MSS = MTU - IP 首部 - TCP 首部 = 1500 - 20 - 20 = 1460 字节**。

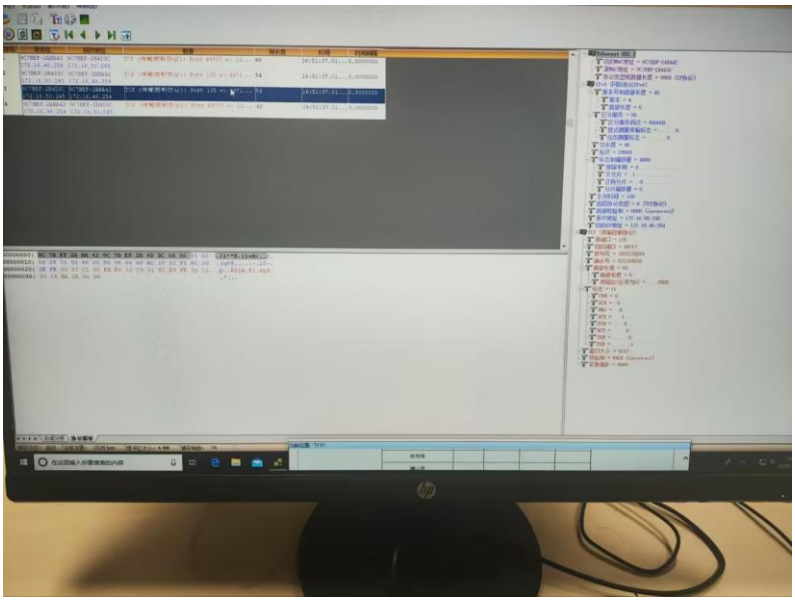
4. 主机 A 断开与主机 B 的 TCP 连接。

5. 查看主机 B 捕获的数据，填写下表。

断开后捕获到的四次“挥手”:







字段名称	报文 4	报文 5	报文 6	报文 7
序列号	831318525	3991278281	3991278281	831318526
确认号	3991278281	831318526	831318526	3991278282
ACK	1	1	1	1
FIN	1	0	1	0

● 结合步骤 3、5 所填的表，理解 TCP 的三次握手建立连接和四次握手的释放连接过程，理解序号、确认号等字段在 TCP 可靠连接中所起的作用。

答案：

- **序号：** 用于给发送的数据字节进行编号，保证接收方能够按照正确的顺序重组数据，并能检测数据是否丢失或重复。
- **确认号：** 期望收到对方下一个报文段的第一个数据字节的序号。确认号表明“该序号之前的所有数据我都已正确接收”，这是实现 TCP 可靠传输（ARQ 机制）的核心，用于触发重传。
- **三次握手与四次挥手：** 三次握手确保了双方都能发送和接收数据，并同步了初始序号；四次挥手则是因为 TCP 是全双工通信，每个方向的连接关闭都需要单独发送 FIN 和确认 ACK，因此通常需要四次交互（有时中间两步会合并）。

三、 实验总结与收获

本次实验利用协议分析与编辑工具，对传输层两大主流协议 UDP 和 TCP 进行了深度的对比分析，实验过程加深了我对网络协议栈设计思想的理解，具体总结如下:

- **“尽力而为”与“可靠交付”的对比体验：** 实验直观地展示了 UDP 和 TCP 的本质区别。UDP 的头部仅 8 字节，无状态、无连接，发送前无需握手，甚至校验和都可以省略（置 0），这种“发射后不管”的模式展现了其极低的开销和极高的传输效率。相反，TCP 为了实现可靠交付，头部至少 20 字节，且必须经历繁琐的三次握手。这种对比让我明白了网络设计中的权衡：在效率与可靠性之间，不同的应用场景需要选择不同的传输层协议。
- **TCP 状态机与滑动窗口机制的验证：** 通过捕捉 TCP 连接建立与释放的数据包，我验证了课本上的 TCP 有限状态机理论。特别是在观察 Seq 和 Ack 的数值变化时，我理解了 TCP 是基于字节流进行编号的。Ack 号总是等于“期望收到的下一个字节的序号”，这一机制是 TCP 实现流量控制（虽然本次实验未涉及窗口大小调整）和差错控制（重传机制）的基石。此外，对 MSS 字段的分析，让我理解了 TCP 层如何根据链路层的 MTU 特性来协商最大报文段长度，避免了 IP 层的分片带来的性能损耗。
- **协议栈封装与校验和机制的细节掌握：** 在手动构建 UDP 包的过程中，我深刻体会到了“封装”的概念。数据从应用层下来，经过 UDP 层加端口、IP 层加地址、MAC 层加物理地址，每一层都独立工作又紧密配合。特别是计算 UDP 校验和时必须包含“伪首部”，这一设计细节让我明白了传输层不仅校验数据本身，还“跨层”校验了源和目的 IP，从而防止了 IP 欺骗或路由错误导致的投递失误。
- **综合实践能力的提升：** 不仅巩固了理论知识，我还提升了实际动手能力。能够熟练

使用抓包软件分析 16 进制数据流，并将其映射为协议字段含义，这对于未来从事网络编程或网络运维工作打下了坚实的基础。